

Rendezés K

Tekintsük a következő 14 elemű számsorozatot, és tegyük növekvő sorrendbe a leírás alapján!

Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Elem	18	23	17	19	25	28	16	21	16	15	23	18	19	20

A rendezés gondlatmenete a következő:

- Menjünk végig az adatsorozaton az első elemtől az utolsó előtti elemig, az elem, ahol tartunk legyen indexelve az **egyik** számmal.
- Minden **egyik** érték mellett menjünk végig a sorozat egyik utáni elemein egy másik indexel, ez legyen a **masik** névű szám.
- Hasonlítsuk össze az **egyik** és a **masik** index által mutatott elemeket, és ha az **egyik** helyen álló érték nagyobb, mint a **masik** helyen álló érték, akkor cseréljük fel őket.

Nézzük lépésenként! Kezdetben **egyik** értéke 0, **masik** értéke 1. Így összehasonlítjuk ezt a két elemet, és nem cserélünk, hiszen $12 < 23$.

Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Elem	18	23	17	19	25	28	16	21	16	15	23	18	19	20

Ezután **masik** értéke 2 lesz, most összehasonlítjuk az a 18-at a 17-tel, s mivel $18 > 17$, ezért cserélünk, vagyis ez lesz az állás:

Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Elem	17	23	18	19	25	28	16	21	16	15	23	18	19	20

Ezután **egyik** értéke tovább nő, **masik** értéke ismét végigfut a sorozat további számain. Cserélni akkor fog, amikor az **egyik** által mutatott szám nagyobb, mint a **masik** által mutatott, tehát két esetben: amikor a 17-et 16-ra cseréli:

Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Elem	16	23	18	19	25	28	17	21	16	15	23	18	19	20

illetve amikor a 16-ot 15-re:

Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Elem	15	23	18	19	25	28	17	21	16	16	23	18	19	20

Az **egyik** mutató nulla értéke mellett már nem lesz több csere.

Most nőjön **egyik** értéke, vagyis 1 lesz: ezután a sorozat további tagjait hasonlítjuk össze vele, tehát először a 23-at a 18-cal:

Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Elem	15	23	18	19	25	28	17	21	16	16	23	18	19	20

Mivel 18 a kisebb, ezért cserélünk:

Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Elem	15	18	23	19	25	28	17	21	16	16	23	18	19	20

Most folytatjuk **masik** növelését, amely végigfut a sorozaton. Az első következő csere akkor lesz, amikor a 18-at a 17-re cseréli:

Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Elem	15	17	23	19	25	28	18	21	16	16	23	18	19	20

A következő csere akkor lesz, amikor egy 17-nél kisebb számot talál, ez az első 16-os:

Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Elem	15	16	23	19	25	28	18	21	17	16	23	18	19	20

A továbbiakban **egyik**=1 értéke mellett nem lesz csere.

Növeljük ismét **egyik** értékét, ami most 2 lesz. A **masik** változó ekkor a 2-es indexű elemet hasonlítja össze az utána következőkkel. Itt már az elején azt találja, hogy $23 > 19$, vagyis cserélni fog.

Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Elem	15	17	19	23	25	28	18	21	16	16	23	18	19	20

A következő csere az első 19-nél kisebb számnál lesz:

Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Elem	15	17	18	23	25	28	19	21	16	16	23	18	19	20

Majd a következő csere a 18-nél kisebb első számnál, ami a 16 lesz történik. A továbbiakban nem lesz csere.

Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Elem	15	17	16	23	25	28	19	21	18	16	23	18	19	20

Ezután **masik** értéke tovább nő, masik értéke végigfut a sorozat további részén, és így tovább ... egészen addig, amíg **egyik** értéke az utolsó előtti szám nem lesz, és végül azt a számot összehasonlítja az utolsóval, és cseréli, ha szükséges. Ezután a sorozat növekvő sorrendbe rendezett lesz.

1. Nyisd meg az órán elkészített Rendezések programot!
2. Készíts egy **RendK()** nevű eljárást, amiben megvalósítod a fent leírt rendezést!
3. A **Rendk()** függvény az órán elkészített rendezéshez hasonlóan adja vissza az összehasonlítások és a cserék számát!
4. A főprogramban hívd meg a most elkészített rendezést egy 30 hosszú számsorozattal, majd írd ki a rendezetlen és a rendezett sorozatot, valamint az összehasonlítások és a cserék számát!
5. A kész munkát **git add . // git commit -m "RendK" // git push** paranccsal töltsd föl a repóba!